REMARKS

Status of this application

Claims 1-18 are pending. Claims 1, 3-5, 10, and 12-14 were rejected under 35 U.S.C. §103(a) as being directed to subject matter deemed to be obvious in view of the combined teachings of U.S. Patent 6,292,880 issued to Mattis et al. (hereinafter "Mattis") and U.S. Patent 6,594,700 issued to Graham et al (hereinafter "Graham"). Claims 2, 6-9, 11, and 15-18 were rejected under 35 U.S.C. §103(a) as being unpatentable over Mattis in view of Graham, and further in view of U.S. Patent Application Publication No. 2002/0099734 filed by Schroeder et al.(hereinafter "Schroeder").

This amendment amends independent claims 1, 6, 8, 10, 15, and 17 to more clearly define applicants' invention.

The Obviousness Rejection

As pointed out by the Examiner in his rejection of claim 1 in Section 3 of the Office Action, Mattis discloses a cache which compares each incoming request message (the name or URL of a requested object) with previously received and stored request messages. Mattis does this by applying a hash function to the object's name as shown in Fig. 3B to form a name key. If a match is found between the resulting name key and a given previously stored name key (that is, if the newly formed name key is found in a directory table), then a stored response (the requested object) is retrieved from the cache and returned to the sender.

As conceded by the Examiner, Mattis does not disclose converting the incoming request message into an incoming canonical request message expressed in a predetermined standard form. However, Examiner states, in Section 3 on page 3 of the Action, that:

> *"Graham discloses converting the incoming request message into an incoming canonical request message (the Office takes the term "canonical" to mean "of or pertaining to a standardized form") 412 expressed in a predetermined standard form (Figure 7, col. 9, lines 10-41). It would be obvious to a person of ordinary skill in the art at the time the invention was made to combine the teaching of Mattis with Graham in order to facilitate the different protocols in the system to work in harmony between the client and server providers, thereby increasing interoperability of the systems with one another."*

Reconsideration is requested.

The Graham system does not convert request messages into canonical form to perform caching Graham's incoming "service requests" are compared with service advertisements placed in a central registry by service providers. In order to serve diverse clients which submit service requests using diverse protocols, each service request is converted into a canonical format so it can be more easily compared with stored service advertisements using a standard comparison mechanism. The mechanism described by Graham in Fig. 7 and at col. 9, lines 10-41, does not compare incoming canonical request messages with previously received and stored canonical request messages to determine if the incoming request message is logically equivalent to said previously received and store canonical request message as required by all of applicants' claims. Instead, Graham converts request messages into a canonical form so that they can be more easily matched against the service advertisements in the internal registry, but not with previously received request messages, and not to determine if the incoming request message is logically equivalent to a previously received request message.

Graham's mechanism for converting incoming messages into canonical form is not related to caching and nothing in Graham suggests modify Mathis' caching mechanism. The Examiner suggests that it "*would be obvious to a person of ordinary skill in the art at the time the invention was made to combine the teaching of Mattis with Graham in order to facilitate the different protocols in the system to work in harmony between the client and server providers, thereby increasing interoperability of the systems with one another.* " But while Graham's service broker mechanism is indeed used to "*allow disparate systems using different protocols the ability to share information easily without the need for mandated data formats*," Mattis' cache is used to match prior requests for information (a name or a URL) with previously received names and URLs. In Mattis' system, all of these incoming names or URLs arrive under a single protocol (HTTP) and Mattis has no need for a system serves a broker between different protocols,. Hence one of ordinary skill in the art would have no reason to modify Mattis caching system to somehow incorporate the Graham protocol broker.

Because the cited references, taken singly or in combination, describe neither the difficulty of efficiently caching data when different requests for that data are logically identical but have different forms, and none of the reference suggests the solution to that problem as claimed by applicants, it is accordingly requested that the rejection of claims 1, 3-5, 10, and 12-

14 as being directed to subject matter deemed obvious in view of the combined teachings of Mattis and Graham be withdrawn.

Applicant has amended independent claims 1, 6, 8, 10, 15 and 17 to further clarify the fact that, in applicants' invention as claimed, incoming request messages are placed in canonical form and then compared to previously stored canonical requests to determine if the incoming request message is logically equivalent to one of the previously received and store canonical request messages, and to return the cached response if a match is found. None of the cited references disclose or suggest this technique.

Applicants' claims 2, 6-9, 11 and 15-18 further specify that all or part of the request messages which are translated into canonical form are expressed in XML. As pointed out in applicants' specification, data requests expressed in XML may be logically identical but have different content; for example, logically identical XML request messages may have different line ending characters or include different whitespace characters which change the form but not the logical meaning of the request. Applicants' claimed technique of converting incoming XML requests into canonical form for storage and comparison permits logically identical requests to be identified even though they don't have identical content as received. Nothing in any of the cited references discloses or suggests doing this.

As the Examiner concedes in Section 8, neither Mattis nor Graham discloses that a portion of the incoming request message be expressed in XML language or that it should be translated into a standard canonical XML form. Neither the caching system described by Mattis nor the protocol broker taught by Graham deal with the special problems associated with caching XML requests.

The Examiner notes that "*Schroeder discloses an incoming data object in XML language that is translated into a standard canonical XML form.*" But the cited passage of Schroeder at paragraphs [0048] and [0049] does not deal with caching, but instead with "normalizing" received XML data objects by passing them through standard XSL transforms. There is no suggestion in the cited passage of Schroeder that these "data objects" are requests or that, once "normalized" that they are stored or compared with previously stored prior requests. In short, there is nothing in the cited passage of Schroeder that suggests that incoming requests should be placed in canonical form so that they can be compared with previously stored canonical requests to identify prior requests that are logically identical to the incoming request as claimed.

Conclusion

Reconsideration of this application and allowance of claims 1-18 as now presented is requested.

Respectfully submitted,

Dated: December 26 2005

Charles G. Call, Reg. 20,406

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this *Amendment* is being transmitted by facsimile to the central facsimile number of the U.S. Patent and Trademark Office, (571) 273-8300, on December 26, 2005.

Dated: December 26, 2005 Signature _____

Charles G. Call, Reg. No. 20,406
USPTO Customer No. 021253
68 Horse Pond Road
West Yarmouth, MA 02673
Ph. (508) 778-2630 Fax (508) 629-6540
call@patentsoft.com